

# SYSTÈMES D'EXPLOITATION

## Devoir Surveillé n°2

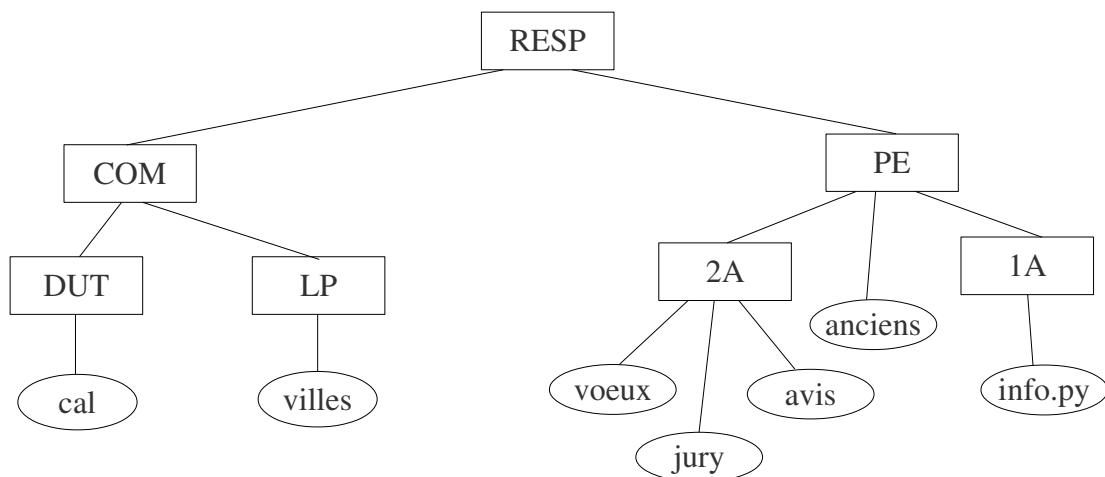
date : 23 novembre 2012 16h45  
feuille A4 manuscrite personnelle autorisée.

durée : 1 heure  
barème indicatif : P1/8, P2/12.

**Lire les questions attentivement et complètement.**

### Partie 1: Protection des fichiers (24 minutes, 8 points)

Voici une partie de l'arborescence du système de fichiers du département informatique.



On s'intéresse à ce que peuvent faire certains utilisateurs de ce système. Voici les noms de ces utilisateurs et les groupes auxquels chacun appartient :

nom	groupe
brigitte	adcom
marie	info, scol
erwann	scol

Voici un ensemble d'informations relatives aux répertoires de cette arborescence (résultat de l'exécution de la commande ls avec les options appropriées).

```
drwxr-xr-x  1 brigitte  adcom  02 nov 21 14:35  COM
d-wxr-xr-x  1 brigitte  adcom  02 sep 26 17:25  PE
drwxrwxrw-  1 erwann    adcom  02 oct 17 15:44  COM/DUT
drw--wxr-x  1 marie     scol   02 nov 28 18:43  COM/LP
d-wxrwxrw-  1 brigitte  info  02 nov 27 13:34  PE/1A
drwx-wxr-x  1 erwann    scol   02 sep 19 17:43  PE/2A
```

Ensuite, voici un ensemble d'informations relatives aux fichiers de cette arborescence

```
-rwx-w-rw- 1 erwann info 02 dec 11 11:10 anciens
-r--rw-r-- 1 brigitte info 02 sep 22 19:20 avis
-rwxr--r-- 1 brigitte scol 02 sep 27 16:53 cal
-rwx-wxr-x 1 erwann scol 02 dec 11 11:10 info.py
-rwx--xrw- 1 marie scol 02 nov 14 18:20 jury
-rwxrw-rw- 1 erwann info 02 sep 16 17:12 villes
-rwxrw-r-x 1 marie adcom 02 dec 13 19:23 vœux
```

On s'intéresse au problème : qui est autorisé à exécuter telle commande dans tel répertoire .

Répondre aux questions directement sur la feuille de réponse de la partie 1, **en respectant rigoureusement les étapes montrées dans l'exemple fictif de la première ligne.**

**On demande d'appliquer les règles du modèle de base vu en cm, td et tp, sans chercher à supposer un 'pseudo' fonctionnement 'logique', ou inventer une hiérarchie des droits selon la catégorie, par exemple : le propriétaire aurait tous les droits de toutes façons !**

## Partie 2: Scripts Bash (36 minutes, 12 points)

**Préparez vos réponses sur papier de brouillon avant de les recopier au propre. Les ratures sales seront pénalisées.**

On voudrait programmer un script appelé **pkg-config** qui affiche les options de compilation qu'il est nécessaire de rajouter pour compiler un programme C lorsqu'on emploie un certain « paquet », par exemple la librairie mathématique. Voici un exemple, **pkg-config math** affiche les options de compilation que demande le paquet « math ». On a juste à les fournir à la commande **cc** :

```
prompt% pkg-config math
-lm
prompt% cc monprog.c -o monprog -lm
prompt%
```

Le principe du script **pkg-config** est de s'appuyer sur un ensemble de fichiers de configuration situés dans **/usr/lib/pkgconfig**. Dans ce dossier, on trouve des fichiers, un par paquet, par exemple **math.pc** qui contiennent plusieurs lignes, dont voici un exemple :

```
prompt% more /usr/lib/pkgconfig/math.pc
desc: librairie mathématique standard : trigo, log...
options: -lm
inclus: math.h stdlib.h
depend: base
prompt% more /usr/lib/pkgconfig/base.pc
desc: options de base pour compiler
inclus: stdio.h stdlib.h
options: -Wall
prompt%
```

On trouve une ligne **desc**: c'est la description du « paquet ». On trouve une ligne **options**: c'est ce qu'il faut afficher à l'utilisateur quand il lance **pkg-config**, par exemple **-lm** pour le paquet **math**. Ensuite, on trouve une ligne **inclus**, c'est la liste des fichiers qu'il faudra inclure par **#include** dans le source c (question 7). La ligne **depend** indique quel autre paquet doit être examiné (question 8).

Les questions sont sur la feuille de réponse n°2. Écrire petit mais lisiblement.